Author(s)

First Name	Middle Name	Surname	Role	Type (Corresp)
David		Casasent	**[title?]**	Y

Affiliation

Organization	URL	Email
Laboratory for Optical Data Processing, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213-3890; phone 412-268-2464; fax: 412-268-6345		casasent@ece.cmu.edu

Author(s)

First Name	Middle Name	Surname	Role	Type (Corresp)
Ashit		Talukder	**[title?]**	

Affiliation

Organization	URL	Email
Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213-3890		

Author(s)

First Name	Middle Name	Surname	Role	Type (Corresp)
Pamela		Keagy	**[title?]**	

Affiliation

Organization	URL	Email
Agricultural Research Service, Albany, Calif. 94710		

Author(s)

First Name	Middle Name	Surname	Role	Type (Corresp)
Thomas		Schatzki	**[title?]**	

Affiliation

Organization	URL	Email
Agricultural Research Service, Albany, Calif. 94710		

Publication Information

Pub ID	Copyright Date	Copyright Org.	Pub Date	Acq No	ISSN
Transactions of the ASAE	© 2001 (check if copyright or not)	American Society of Agricultural Engineers	Vol. 44(2): xxx-xxx	FPE 1496	0001-2351

DETECTION AND SEGMENTATION OF ITEMS IN X-RAY IMAGERY

D. Casasent, A. Talukder, P. Keagy, T. Schatzki

Article was submitted for review in June 1998; approved for publication by the Food & Process Engineering Institute of ASAE in September 2000. Presented in parts as Papers No. 2907-21 and 3073-11 of the Society of Photo-Optical Instrumentation Engineers. Approved for publication by Agricultural Research Service, Western Regional Office, July 1997.

The authors are **David Casasent, [title?]**, **Ashit Talukder, [title?]**, Laboratory for Optical Data Processing, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh; and **Pamela Keagy, [title?]**, **Thomas Schatzki, [title?]**, Agricultural Research Service, Albany, California. **Corresponding author:** David Casasent, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213-3890; phone: 412-268-2464; fax: 412-268-6345; e-mail: casasent@ece.cmu.edu.

Abstract. *Processing of real-time X-ray images of randomly oriented and touching pistachio nuts for product inspection is considered. Processing to isolate individual nuts (segmentation) is emphasized. The processing consisted of a blob coloring algorithm, filters, and watershed transforms to segment touching nuts, and morphological processing to produce an image of only the nutmeat. Each operation is detailed and quantitative data for each are presented. These techniques are useful for many different product inspection problems in agriculture and other areas.*

Keywords*. Blob coloring, Image processing, Morphological processing, Watershed transform, Segmentation, Detection, X-ray sensors.*

X-ray sensors have proven useful for many agricultural product inspection applications. These include detection of contaminants of high atomic number (Schatzki and Wong, 1989), interior heart holes in potatoes (Finney and Norris, 1978), weevil damage in wheat (Keagy and Schatzki, 1993; Milner et al., 1950), worm damage in navel oranges (Keagy et al., 1996), and subtle differences such as local regions of water in apples (Tollner et al., 1992; Schatzki et al., 1997) and lettuce (Schatzki et al., 1981a,b). X-ray images of pistachio nuts are considered in our case study. The internal product detail that X-ray images provide allows the presence of worm damage and other defects to be determined by non-destructive (non-invasive) methods. This is of concern, since worms contribute to conditions favoring mold growth and aflatoxin production, and their presence must be reduced (aflatoxin concentration must be below 4 ng/g for import to a number of countries). Standard inspection techniques cannot determine many defects that are possible with X-ray images. Prior work (Keagy et al., 1996; Sim et al., 1996) confirmed that X-ray image processing can identify infested versus good pistachio nuts. However, this work did not address the problem of touching nuts, and sometimes required that all nuts be in the same orientation. It also did not employ a large database.

In this article, new image-processing algorithms are considered that segment each nut independent of its orientation. New image-processing methods were used to produce images of only the nutmeat. This omitted gray-scale textured data in the air gap between the nutmeat and shell, which can disrupt an image-processing system that needs to analyze the gray level in only the nutmeat. This article emphasizes the image-processing algorithms used to produce data files of individual nuts (segmentation) and to produce nutmeat-only images. The new image-processing techniques described in this article are of use in processing many other agricultural products for inspection.

The database considered in this work consisted of 29 trays of pistachio nuts. Twenty-five of the trays were of large nuts (18-20 nuts/oz). Two trays of medium nuts (21-25 nuts/oz) and two trays of small nuts (26-30 nuts/oz) were also included to show the generality of the image-processing algorithms. An image of each tray was obtained.

It contained about 100 nuts in random orientations, with an average of nine clusters of two to four nuts touching in each image. The nuts were obtained from a processor after sorting and sizing and after hand-inspection to remove twigs and other non-nut material. These are typical images of such products on a conveyor belt that would be input to an automatic vision inspection system

MATERIALS AND METHODS

This section describes the database used and notes the problems and issues associated with segmentation. Basic morphological image-processing methods are reviewed, and several prior segmentation algorithms are considered.

X-RAY IMAGES

Each tray was X-rayed for 90 s at 25 keV (with an 0.25 mm Be window) using a Faxitron series X-ray system (model 4380N, Faxitron Corp., Buffalo Grove, Illinois) and Industrex B film (Eastman Kodak, Rochester, New York). Twelve-bit digital images of these X-ray films were obtained at a resolution of $(0.173\text{ mm})^2$/pixel using a Lumiscan 150 film scanner (Lumisys, Sunnyvale, Calif.). Each film included a 25-step plastic wedge constructed to provide absorption levels comparable to those found in nuts. Small variations between films, due to exposure and processing, were removed by linearly rescaling the gray values of each film to constant values in the step wedge. The optical density response of the film was sigmoidal when plotted against the thickness of the step wedge. The variance of the optical density was an inverse function of the optical density. A logit transform was used to linearize the response and equalize the variance. By pixel averaging, these images were reduced in pixel count by a factor of nine, to produce 12-bit images with a resolution of $(0.5\text{ mm})^2$/pixel. These will be referred to as "X-ray film images."

A set of real-time X-ray linescan images of these nuts was also obtained. A custom-built, line-scanning, moving-belt unit (EG&G Astrophysics, Harbor City, Calif.) was used. This prototype food X-ray system used a 0.015 cm-thick aluminum filter and a stabilized 40 keV source delivering 18 mA. A collimated, fan-shaped X-ray beam passed through the samples onto an 800-element linear detector array with a pitch of 0.5 mm/detector. The detector array was overlaid with a $Gd_2\ O_{3-x}\ Sx$ (Tb-doped) plastic-imbedded phosphor (Tri-Max 12, 3M Corp., St. Paul, Minn.) that converted X-ray photons to green light to activate the photodiodes. The line scanning rate was 120 Hz (corresponding to a belt rate of 60 mm/s). The resultant 12-bit images have the same $(0.5\text{ mm})^2$/pixel resolution as the film images. These will be referred to as "linescan images."

IMAGE PROCESSING OBJECTIVES

Both X-ray film and linescan images of all 29 trays were used in the tests. Figure 1a shows a typical tray of nuts, and Figure 1b shows X-ray images of several nuts in which internal detail useful for detecting worm infestation and for classification is apparent. The individual nut images also indicate some of the image processing problems. Specifically, each nut consists of a nutmeat region, an air gap, and an outer shell region. The intensity within the nutmeat region varies. This region contains dark areas due to blemishes, worm tunnels, and nutmeat splits. In addition, the air gap has lower pixel values, the shell region contains higher pixel values, and the width of the air gap varies. These factors complicate extracting an image of only the nutmeat region, which is the input to the feature extractor and classifier. Prior vision work on pistachio nuts (Keagy et al., 1996) included images of the entire nut (nutmeat, gap, and shell).

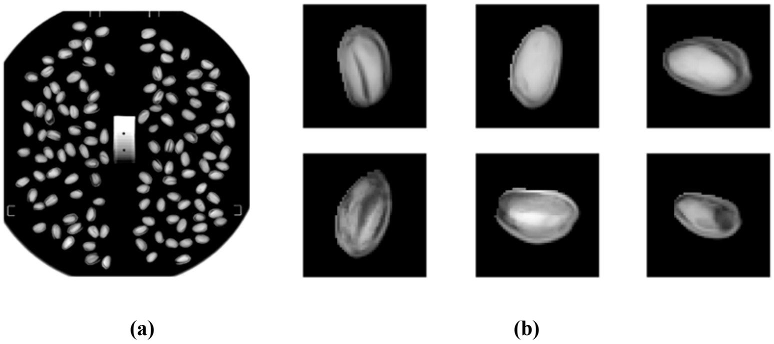


Figure 1. Typical scanned X-ray image of a tray of pistachio nuts (a) and individual nuts (b) showing internal detail.

Before an image of only the nutmeat can be obtained, segmentation is necessary; i.e., a separate image file of each nut must be obtained. Within each image, there are an average of nine clusters of touching and overlapping nuts. Figures 2a and 2b show two typical images of nut clusters. As can be seen, nuts can touch in various ways and result in large overlapping regions. After binarization, holes will be present on some nuts (fig. 2a). In addition, the size of large nuts can vary significantly, from 20 to 25 pixels in width and from 25 to 40 pixels in length (380-900 pixels per nut), as seen by comparing figure 2a with figure 2b. Linescan images (fig. 2d) have lower contrast, more dark regions, and more noise than X-ray film images (fig. 2c). In addition, each nut in a linescan image is about 6% larger due to the fan angle of the X-ray beam. Thus, there is more of an overlapping touching region in linescan images than in X-ray film images (fig. 2d vs. fig. 2c). These factors complicate segmentation. The following sections review some of the conventional image processing methods relevant to this application.

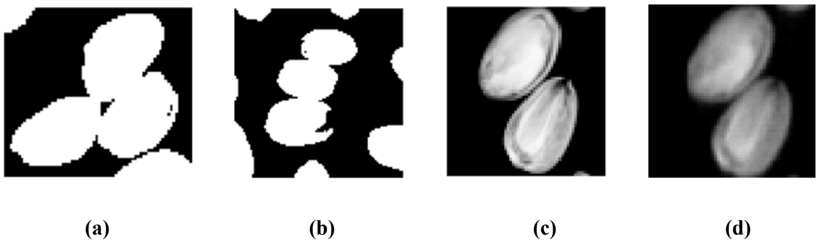


Figure 2. Binary images of typical nut clusters (a) and (b), and typical X-ray film (c) and linescan (d) image differences.

Morphological Processing

The morphological erosion (dilation) of a binary image is achieved by correlating the image with a structuring element (SE) filter and thresholding the output at a threshold T. The SE is typically a disk. The threshold T is one or the number of pixels in the SE for a dilation or erosion, respectively. Erosion removes any white object regions smaller than the size of the SE and reduces the boundary of large white regions by half the width of the SE. A morphological opening is an erosion followed by a dilation. It removes white areas smaller than the SE surrounded by black pixels and does not affect white areas larger than the SE.

Gray Watershed Algorithm

This standard algorithm is useful for segmenting an object from surrounding clutter. It was applied to the gray-scale input image (a nut cluster) in the present case. Figure 3 depicts the steps in the algorithm. The center of each nut is assumed to have the highest pixel value, with lower values at the nut edges. This image was then complemented so that nut centers now have the lowest values. Figure 3a shows a hypothetical complemented 1-D image of a cluster of two nuts. It has basins (minima) at the center of each nut object. The algorithm is best described by envisioning filling this contour with water. This results in the different basins being filled to equal heights. When the water overflows between two adjacent basins, the corresponding ridges or watersheds locate the dividing lines between basins, or in the current problem, the boundary between adjacent nuts (each nut corresponds to a separate basin). This intuitive explanation (Vincent and Soille, 1991), a description of the algorithm (Dougherty, 1986), and the associated mathematical details (Dobrin et al., 1994; Viero et al., 1994) are available elsewhere.

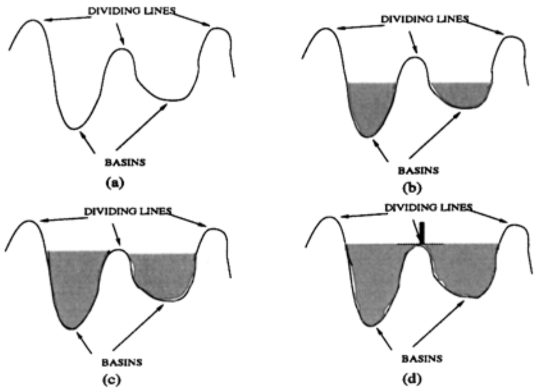


Figure 3. Intuitive explanation of the gray watershed algorithm showing a complemented image (a) and different levels of water in the basins (b to d).

BINARY WATERSHED ALGORITHM

This algorithm (Serra and Vincent, 1989) is another segmentation algorithm similar to the gray watershed algorithm, and it has many variations. It can be described in terms of erosions (Dougherty, 1986), but it is best implemented using the distance transform (Serra and Vincent, 1989). In this case, the distance transform of a binary nut cluster is calculated and the gray watershed algorithm is applied to the gray-scale distance-transformed image. This combination is referred to as the binary watershed algorithm. To produce the distance-transformed image, the nut cluster image was thresholded to obtain a binary blob. Each "on" pixel in a nut was replaced by a gray-scale value equal to its distance from the nearest edge in the nut cluster boundary. Figure 4b shows an example of the distance transform image of figure 4a. Its cross-section (fig. 4c) shows two well-defined peaks and is much better than the cross-section produced from the original gray-scale image. Figure 4d shows the complement of the cross-sectional plot in figure 4c. Note the prominent valleys (centers of the nuts) and the peak (the dividing line between the nuts). The gray watershed transformation of figure 4d yielded the final segmented image.

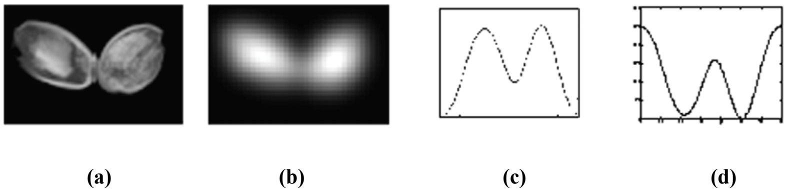


Figure 4. 1-D example of basins produced for a nut cluster (a), using the distance transform (b), shown in cross-section (c), and complemented (d).

IMAGE PROCESSING PROCEDURES

OVERVIEW OF THE PROCESSOR

Figure 5 shows the block diagram of the image processor developed for this study. Thresholding produces a binary image. Blob coloring produces lists of connected "on" pixels. These include individual nuts, nut clusters, and small noise pixel regions. Each connected blob region is given a different gray-scale (color). Nut clusters are subsequently segmented to locate each individual nut. As will be described later, this requires an estimate of the number of nuts per cluster and the center of each nut in each cluster. Morphological processing is then used to produce an image of only the nutmeat portion of each nut image. These different steps are described in the following sections.

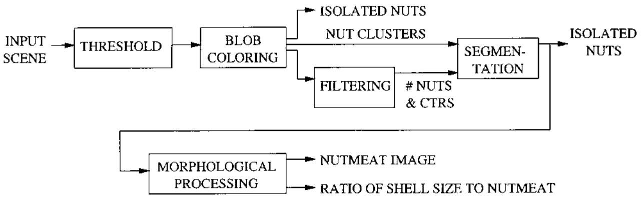


Figure 5. Image processor block diagram.

THRESHOLDING

Each tray image was first binarized. This produced binary data files for faster processing. Pixels with gray values above a threshold are 1 (white), and pixels below the threshold are 0 (black). In the X-ray film images, the background was zero, and the darkest portion of a nut was the gap (gray level ≥ 20) between the nutmeat and the shell. Thus, any threshold between 0 and 20 was sufficient to binarize the present data. For this study, a threshold of 15 was used. The linescan images had a non-zero background level, between gray levels of 0 and 10, with the darkest portion of a nut having gray level 25. The same threshold of 15 was used. These thresholds produced binary images with almost all pixels within each nut being "1" (figs. 2a and 2b). Later, all pixels within isolated nut images will be filled in to avoid possible small holes ("0" valued pixels) within a nut.

BLOB COLORING

After thresholding, the tray image contained a number of isolated nuts and several clusters of touching nuts (figs. 2a and 2b). The tray image (fig. 1a) was blob colored to locate these. Each blob contained all connected pixels in a region of the input image. Each blob was assigned a different gray level (color) and could contain one nut or several touching nuts.

The blob coloring routine used was a fast, efficient implementation of a standard algorithm (Ballard and Brown, 1982). The binary tray image was lexicographically scanned, each connected region or blob was assigned a unique gray-level (color), and the blobs (nuts) were run length encoded and stored in a list. The algorithm used was as follows:

1. Rows of the image are scanned until an "on" pixel is encountered. This starting position, and the number of consecutive "on" pixels in the first row of the blob, are stored. This is a form of entropy coding known as run length encoding.
2. After a run/scan has ended, the current row is scanned until another "on" pixel is encountered. A new run length at this pixel is started and a new color (blob) is assigned to this new run of "on" pixels.
3. The next row is then scanned to produce run lengths for it while monitoring the run length data from the prior row. When "on" pixels in both rows have the same column, then their runs belong to the same blob, and the run on the present row is assigned the same color as the run in the prior row. Processing continues in this manner, looking for new runs until the end of the row is reached. This converts all prior parts of one blob to the same color. Discarded colors go back into the list of available colors.
4. This process is repeated for each row in the image. There may be more than one run in each row. The algorithm colors touching nuts and groups them into one file. If there is a hole ("off" pixel) in a nut, the algorithm still colors all "on" pixels in it and connecting nuts to the same color.

NUMBER OF NUTS PER CLUSTER (FILTERING)

After blob coloring, the files contained the "on" pixels in each blob. Several different segmentation methods were considered. The technique chosen required estimates of the number of nuts per blob or cluster and estimates of the center of each nut. The number of pixels in each blob was easily obtained from the sum of the lengths of its runs. The number of "on" pixels in each blob was used to determine if it contained one nut or if it was a cluster of nuts. If a blob contained too few pixels (<350), it was omitted as noise. If a blob contained ≥350 and ≤700 pixels, it was considered to contain one nut. That file was transferred as an isolated nut to our feature extractor. If a blob contained

>700 pixels, it was denoted as a cluster of nuts and was separately processed by the segmentation unit. Because of the large range of sizes of large nuts, the number of nuts in a blob was not easily determined. For this database, most nuts contain 600-800 pixels. However, outliers can have as few as 380 or as many as 900 pixels. Thus, from the number of pixels per cluster, one cannot determine the number of nuts in a blob, or if a blob is a single nut. With a threshold of 1000 used to determine if a blob contained more than one nut, two errors occurred in the 26 trays of large nuts; i.e., two clusters of two nuts were classified as single nuts. Simple estimates of the number of nuts per blob are thus possible, but are not expected to have high accuracy.

Several new algorithms were considered to obtain very accurate estimates of the number of nuts per cluster and the centers of each nut. Morphological erosion of the binary blob cluster images was considered to obtain separate blobs associated with each nut. Since the width of the smallest nut can be 20 pixels, which was less than the length over which two nuts can touch (over 22 pixels), this method failed on many nut clusters. In addition, erosion will grow any holes present on a nut. Thus, erosion was not pursued. Rotation-invariant and distortion-invariant filters (Casasent and Cox, 1998 **[correct citation?]**) were developed. These performed very well, but a Gaussian filter performed slightly better (by a fraction of a percent) and was thus used.

The technique chosen was to correlate the gray-scale image of each blob cluster with a Gaussian filter: $g(x,y) = 1/(2\pi\sigma_x\sigma_y)\, e^{-(x^2/2\sigma_x^2 + y^2/2\sigma_y^2)}$ with variance $\sigma^2 = \sigma_x^2 = \sigma_y^2$. The filtered correlation plane output was a smoothed version of the input with gray-scale variations (due to split nuts, worm tunnels, gaps between the shell and nut meat, etc.) reduced; i.e., in the smoothed output, one peak near the center of each nut in a cluster was expected. The variance chosen was $\sigma^2 = 15$, or $\sigma \simeq 4$. Since the Gaussian width was 2σ at $\simeq$ 70% of the peak, splits, tunnels, or gaps with widths of 8 pixels (the largest found in the database) were smoothed out**. [Disk copy contains brief additional text here].**

The Gaussian filtered output was normalized for each nut cluster to have a maximum gray-level peak of 255. To detect the number of peaks in a Gaussian filtered image, a peak sorting and masking algorithm (Casasent and Ye, 1997) was used. In this algorithm, the Gaussian filtered output plane was thresholded at decreasing gray values until a peak was located. When a peak was located, its location was recorded, and a 15 × 15-pixel window region around the peak was omitted from future processing. The threshold was then lowered until another peak was located, and the process was repeated until a threshold of 100 was reached. The 15 × 15 window removed most of one nut but did not alter the center or peak of adjacent nuts. The minimum threshold was determined from tests on several nut clusters.

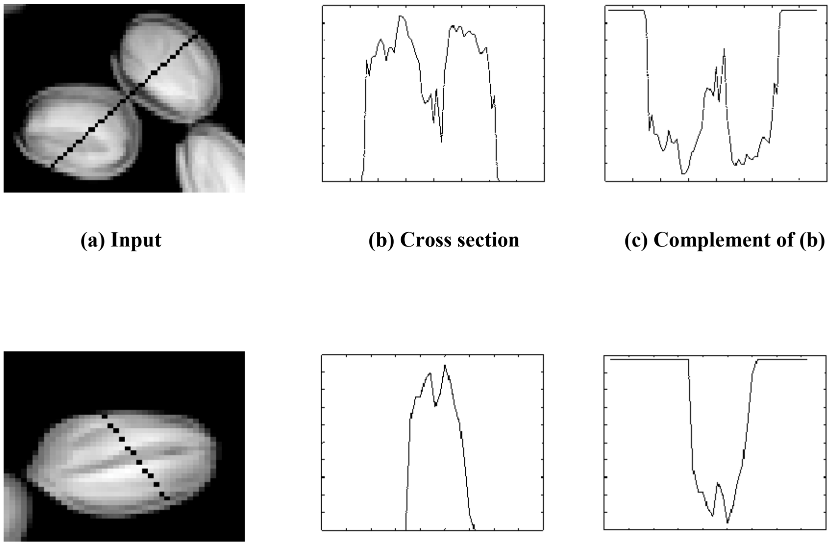
For linescan images, the nut cluster image was eroded with a disk of diameter 13 to reduce the extent over which adjacent nuts touched. Gaussian filtering was then applied to this eroded image. The erosion of the binary nut cluster image removed six pixels around the border of the cluster. This eroded binary image was then used as a mask for the gray-scale nut cluster image; only those pixels present in the binary mask were present in the final eroded gray-scale output cluster image. Gaussian filtering was then applied to this image. Since the erosion can grow holes within a nut image, the binary cluster mask image was morphologically closed with a disk of diameter 3 pixels prior to the erosion; this filled in small holes on the nut.

SEGMENTATION

Several different segmentation methods were considered. To demonstrate the difficulty of the problem, some of the unsuccessful methods are briefly noted here with the reasons for their poor performance. The use of B-splines was considered to describe the boundary of a nut cluster, and the second derivative of the cluster boundary was expected to be large where two nuts meet. This approach assumed that the nut boundaries were smooth so that the boundary regions with high second derivatives corresponded only to points where two nuts touched. Since all nutshells did not have smooth boundaries (see the bottom nut in fig. 2b), this method often failed, and it was not used.

Euclidean and Mahalanobis distance segmentation techniques were also implemented. In these techniques, the distances of all "on" pixels in a cluster to the estimated centers ($\underline{X}_1$, $\underline{X}_2$, etc.) of each nut were calculated. Each pixel in the cluster was assigned to the nut to whose center it was closest. If the nuts in a cluster differed widely in size, had different orientations, or if the estimates of the centers of the nuts were poor, this method failed.

The gray watershed algorithm worked well if the objects to be segmented had slowly varying continuous gray levels and if they did not overlap. Our nut images had large gray-scale variations within the nuts, gaps between the shell and nutmeat, and they touched or overlapped. For touching or overlapping nuts, the gray level in the boundary region (shell) was darker than in the nutmeat region. Thus, a dividing line existed between nuts in the complemented image. The gaps by the shells in these objects caused problems. Local minima occurred in the gaps (as shown in the 1-D gray-scale plot in fig. 6b). These resulted in extra basins on each side of the true nut boundary in the complemented image (fig. 6c). Similarly, worm tunnels and split nuts produced other false dividing lines (figs. 6d-6f). In these cases, the minimum of a basin did not occur at the center of a nut. Individual basins for each object must thus be determined (the nut center estimates allow this and determine the number of basins to consider) to overcome the problems in figures 6d-6e, and techniques to overcome false boundaries larger than the correct boundary (figs. 6a-6c) must be devised.



(a) Input **(b) Cross section** **(c) Complement of (b)**

(d) Input **(e) Cross section** **(f) Complement of (e)**

Figure 6. Cross-section nut scans showing local minima due to nutmeat gaps (a-c) and due to worm tunnels and split nuts (d-e).

Smoothing the image with a Gaussian filter to reduce the effect of local peaks and valleys (to produce smooth images of each object) was considered, but boundary errors still occurred. When gray-level variations were present in an object, the gray watershed algorithm resulted in over-segmentation (one object was separated into several regions), as noted by Meyer and Beucher (1990) and Dougherty (1986). A Gaussian smoothing function with a large kernel has been used, but under-segmentation still occurred (Meyer and Beucher, 1990). Thus, the gray watershed was not suitable for this application, in which objects touched and had internal gray-scale variations, such as gaps, near their edges. It was suitable for removing clutter around an object when the object did not have internal gray-level variations.

The standard binary watershed algorithm had over-segmentation problems if the boundary was irregular or complex (Russ, 1990; Orbert et al., 1993). This problem also arose (Orbert et al., 1993) in processing cashew nut images. The problem is shown in figure 7 for a nut with an irregularly shaped boundary. The distance transform image is shown in figure 7b. The complement of its cross-sectional plot has a minimum at the center of the nut and a local minima on the irregular nut extension (fig. 7c). Multiple basins and boundary lines were also found (Orbert et al., 1993) when the objects were not oval or elliptical (this occurs for some pistachio nuts). Thus, the standard binary watershed algorithm did not work on objects with irregular boundaries.

The segmentation algorithm developed for this study used the binary watershed algorithm with estimates of the number of nuts, and the centers of each nut were used to initiate and define the basins.

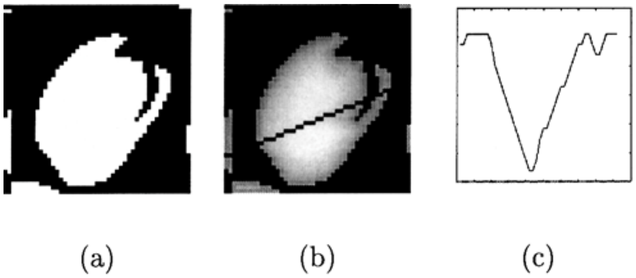


Figure 7. An example of a local false basin produced (c) in the distance transform (b) of an object (a) with an irregular boundary.

MORPHOLOGICAL PROCESSING

The process necessary to produce a final image of only the nutmeat involved a number of morphological processing steps. Once each nut had been segmented, the binary blob colored image defined the set of pixels in the tray image that represented each nut. These binary masks were used to pass the associated gray-scale tray image pixel values to obtain a gray-scale image of each individual nut. The binary mask for each nut must not have holes in it, as this will result in zero-valued pixels in the gray-scale nut image, and erosions will enlarge these holes. To remove holes within the mask for an isolated nut, the mask image was first complemented and blob colored. Next, an image of only the blob due to the background was formed and complemented. This produced an image of the nut with any holes filled in. This cannot be done for a nut cluster blob, as a central hole can correspond to background (fig. 2a). Rather, it was only applied to isolated nuts after segmentation.

To produce an image of only the nutmeat portion of the nut, the edge of the shell and the gap region were removed. A binary erosion was first performed on the binary filled-in mask for each isolated nut with an SE disk of diameter 7. This involved performing a correlation of the binary image with the disk, followed by a threshold equal to the area of the SE. This removed 3 pixels from the outer border of the mask; this was the smallest width of the shell in any nut image. The gray-level pixels in the input that were within this eroded mask were passed. A gray-scale opening with an SE disk of size 5×5 was used to remove white regions smaller than the SE surrounded by dark regions (i.e., any remaining shell surrounded by background and gap). The resultant image was then thresholded midway between the minimum and average gray values in the nut image. This removed the gap. This binary image was used as a mask to pass a reduced number of nut pixels in the original gray-scale nut image, to produce the final gray-scale image of only the nutmeat.

RESULTS AND ANALYSIS

NUMBER OF NUTS PER CLUSTER AND NUT CENTER ESTIMATES

Figure 8 shows a typical result of Gaussian filtering to locate all nuts in a cluster. Figure 8a shows a cluster of three nuts and a separate nut. This figure and the isometric view of this image in figure 8b show that the nuts did not have clear single peaks for each nut due to the nutmeat split and dark nutmeat regions. The Gaussian filtered output in figure 8c shows clear single peaks for each nut.

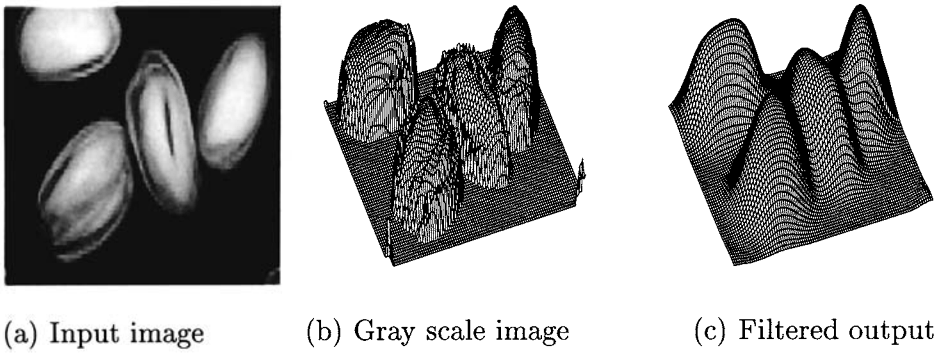


Figure 8. Gaussian filtering to detect nut centers (c) in an input image (a) with gray-scale (b).

The tests on all X-ray film images yielded perfect detection of all nuts in all clusters (detection probability P_D = 100%), including 300 nut clusters of small, medium, and large nuts. For 340 out of the 350 nut clusters in the linescan images, all nuts were correctly located. Only 20 nuts out of 2900 nuts were incorrectly located, thus P_D = 99.3%. An analysis of the errors showed that the reason for the slightly lower P_D for linescan images was due to the larger size of these nuts. This resulted in more nut clusters in linescan images than in film images (350 in linescan images vs. 300 in film images) and a larger region of overlap on touching nuts.

Figures 9a and 9b show binary versions of the film and linescan images in figures 2c and 2d. The larger region of overlap is apparent in figure 9b. This larger area of overlap and the increased amount of dark regions in the linescan images (especially those near the edges of nuts) resulted in lower and less well-defined peaks, which were shifted toward the region of overlap between touching nuts. The algorithm was thus modified by eroding the binary nut cluster images with a disk of diameter 13. This eroded binary image was used as a mask, and only the gray-level input pixels in this binary mask region were retained. The Gaussian filter was applied to this processed image. Since the erosion can grow holes within a nut, the binary cluster image was closed with a disk of diameter 3 prior to the erosion. This fills in small holes in a nut (e.g., the right nut in fig. 2a). With these modifications, P_D = 100% was still obtained on the film images, and an improved P_D = 99.93% was obtained for the linescan images (two clusters or two nuts had errors out of 2900 nuts).

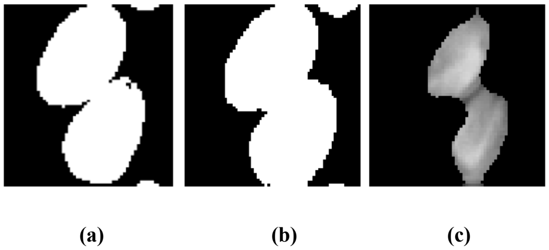


Figure 9. Binary clusters of film (a) and linescan (b) nut images and the resultant region of (b) used after erosion (c).

SEGMENTATION

The algorithm used was the binary watershed with the number of nuts (the number of basins) and the nut centers (and hence which basins are correct) known. The gray watershed algorithm was applied to the distance-transformed image until two separate correct basins merged. Figure 10 shows typical binary watershed segmentation results. The binary watershed correctly segmented 100% of all nuts and nut clusters in the X-ray film images and 99.3% of the nuts and clusters in the linescan images (100% correct segmentation of the 4 small and medium linescan nut images, and 99.2% correct segmentation of the 2500 large nut images). Nuts in 340 clusters out of a total of 350 nut clusters were correctly segmented, and only 20 nuts out of $\simeq$ 2900 were incorrectly segmented.

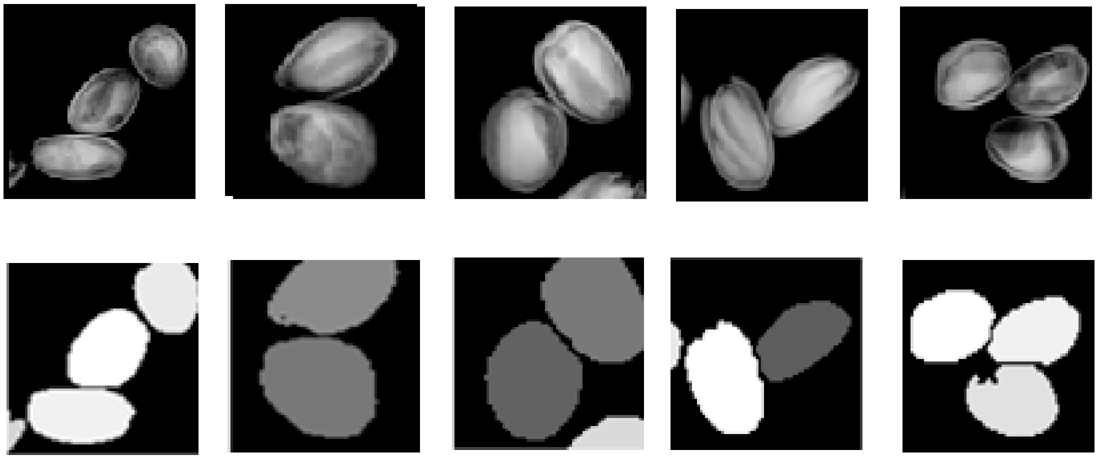


Figure 10. Typical input (top) and binary watershed segmented results (bottom) for different nut clusters.

FINAL NUT MASKS AND SEGMENTED NUT IMAGES

Figure 11 shows an example of the final binary mask used to pass the associated gray-scale image pixels for an individual nut. The original binary mask after segmentation, blob coloring, and thresholding can have holes in it (e.g., fig. 11a). These holes must be filled in, since an erosion will enlarge them. The complement of this mask (shown in fig. 11b) was blob colored, an image of only the blob due to the background was formed and complemented to produce the final nut mask with any holes filled in (e.g., fig. 11c).

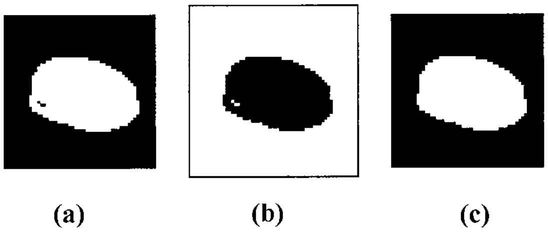


Figure 11. Algorithm to fill in a hole on a nut mask (a) by forming the complement of the image (b) and then using morphological processing to produce the result in (c).

NUTMEAT-ONLY IMAGES

The results of extracting images of only the nutmeat (fig. 12) are discussed here. Figure 12a shows an original nut image. The results after erosion with a disk of diameter 7 to remove the shell are shown in figure 12b. Figure 12b still contains the gap region between the nutmeat and the shell, and it still contains a thin shell region on the right. The result of opening figure 12b with an SE disk of size 5 × 5 is shown in figure 12c. The opening is seen to remove any remaining shell strips smaller than 5 pixels in width. Note that other nut boundaries are not altered. Specifically, the opening will not affect nuts that have no air gap left or nuts with no white shell regions remaining. Nut image regions that do not contain any small white regions smaller than the SE are not modified by the opening. Thus, nutmeat regions with no shell left will not be affected. The nut image in figure 12c was thresholded at a gray level midway between the minimum and the average gray values in the nut image. This removes the air gap and produces the binary mask image in figure 12d. This binary mask (after filling in holes as noted earlier) is then used to pass the associated reduced number of nut pixels in the original image of figure 12a, to produce the final gray-scale image of only the nutmeat in figure 12e.

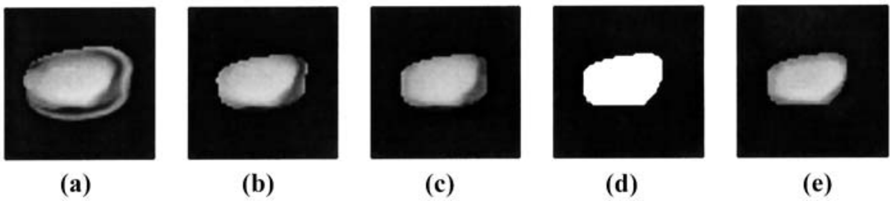


Figure 12. An example of the morphological steps used to obtain a nutmeat-only image (e) from an input image (a) including: erosion (b), gray-scale opening (c), thresholding (d), and passing image through the mask in (d) to produce the result in (e).

FINAL RESULTS

The accuracy of the nutmeat-only image extraction is now quantified. Two errors can occur: under-segmentation (the air gap is not completely removed) and over-segmentation (some of the nutmeat pixels are removed). The features used for classification included the histograms of the nutmeat-only images. Thus, small nutmeat segmentation errors of several pixels were not significant if they occurred on severely infested nuts.

The X-ray film images were considered first. For all good (non-infested) nuts, no significant segmentation errors occurred. When under-segmentation occurred, no more than 3-4 pixels of the air gap remained. This occurred in only 7% of all nuts. In all cases, these dark gap regions did not have pixel values as dark as in infested regions, and the number of gap pixels left was less than 0.5% of the total number of pixels on an average-sized nut. Thus, under-segmentation was not a concern. Over-segmentation only occurred in a few severely infested nuts, since some nutmeats had rather dark gray levels near the nut edge. For 1.3% of the infested nuts (9 out of the 700 infested nuts in the database), over-segmentation occurred. As much as a 4 pixel thick region at the edge of the nutmeat was removed, but only in one or at most 2 isolated regions, not around the entire nutmeat region. These few over-segmentation errors were also of no concern, since they only occurred in severely infested nuts (the rest of the nutmeat had significant gray-scale variations present), and they represented a small percent of the total nutmeat pixels.

For the linescan images, worse accuracy was expected because of the lower contrast and more dark levels present. Out of 2200 good nuts, 50 were under-segmented with, at most, 7 pixels of the gap present on any one nut. These non-removed gap pixel values were not very dark. They represented a small fraction of all nutmeat pixels and were thus not expected to present problems. For the 700 infested nuts, 9 were found to be under-segmented (with at most 8 air gap pixels remaining on any one nut) and only 5 nuts were found to be over-segmented. For reasons noted above, these segmentation errors did not appear to represent significant problems.

SUMMARY AND CONCLUSIONS

A number of standard and new image processing operations have been described to perform segmentation of overlapping and touching objects. These included a fast blob-coloring algorithm, combined with a new binary watershed algorithm. The watershed algorithm was new in its use of the estimates of object centers to initiate and define basins, thus overcoming internal gray-scale object variations. Use of morphological image processing yielded filled-in masks of each object independent of its orientation. The algorithm worked for touching and overlapping objects. These operations (and modifications of them) are useful for many applications in agricultural inspection and other areas. For the specific pistachio nut inspection problem, new morphological image processing operations were advanced to produce gray-scale images of only the nutmeat. These images are expected to produce better features. Initial segmentation results and nutmeat-only extraction on a large database of real-time linescan images of different-sized nuts were excellent:

1. Segmentation of 99.3% of all nuts was achieved.
2. Only 0.25% of the good nuts were slightly under-segmented after nutmeat extraction.
3. Only 0.7% of the infested nuts were over-segmented after nutmeat extraction.

It is expected that these minor nutmeat-only errors will not noticeably affect classification. Future classification tests will address this issue, as well as the detectability expected for infested nuts with present real-time X-ray imaging systems.

Acknowledgements. This material is based upon work supported by the Cooperative State Research, Education, and Extension Service of the U.S. Department of Agriculture, under agreement No. 96-3550-3455. The authors thank David Weber and Westley Cox for contributions to the development of the software used.

REFERENCES

Ballard, D., and C. M. Brown. 1982. *Computer Vision.* Englewood Cliffs, N.J.: Prentice Hall.

Casasent, D., and W. Cox. 1998. RI-MINACE filters to augment segmentation of touching objects. *Pattern Recognition* 34(9): 1311-1317.

Casasent, D., and A. Ye. 1997. Detection filters and algorithm fusion for ATR. *IEEE Transactions on Image Processing* 6(1): 114-125.

Dobrin, B. P., T. Viero, and M. Gabbouj. 1994. Fast watershed algorithms: Analysis and extensions. In *Proceedings of the Society for Photo-Optical Instrumentation Engineers*, 209-220. **[what is "2180?"]**. Bellingham, Wash.: **[publisher's name?]**

Dougherty, E. 1986. An introduction to morphological image processing. In *Tutorial Texts in Optical Engineering, Proc. SPIE*, **[page numbers?]**. **[what is "TT9"?]**. D. C. O'Shea, ed. Bellingham, Wash.: **[publisher's name?]**.

Finney, E., and K. Norris. 1978. X-ray scans for detecting hollow heart in potatoes. *American Potato Journal* 55(**[issue no.?]**): 95-105.

Keagy, P. M., B. Parvin, and T. Schatzki. 1996. Machine recognition of navel worm damage in x-ray images of pistachio nuts. *Lebensm-Wiss. U. Technol.* 29(**[issue no.?]**): 140-145.

Keagy, P. M., and T. Schatzki. 1993. Machine recognition of weevil damage in wheat radiographs. *Cereal Chemistry* 70(6): 696-700.

Meyer, F., and S. Beucher. 1990. Morphological segmentation. *Journal of Visual Communication and Image Representation* 1(1): 21-46.

Milner, M., M. Lee, and R. Katz. 1950. Application of x-ray technique to the detection of internal insect infestation of grain. *J. Econ. Entomol.* 43(**[issue no.?]**): 933.

Orbert, C. L., E. W. Bengtsson, and B. G. Nordin. 1993. Watershed segmentation of binary images using distance transformations. In *Proceedings of the Society for Photo-Optical Instrumentation Engineers*, 159-170. **[what is "1902"?]** Bellingham, Wash.: **[publisher's name?]**.

Russ, J. C. 1990. *Computer-Assisted Microscopy.* New York, N.Y.: Plenum Press.

Schatzki, T., R. Haff, R. Young, I. Can, L. Le, and N. Toyofuku. 1997. Defect detection in apples by means of x-ray imaging. *Trans. ASAE* 40(5): 1407-1415.

Schatzki, T., S. Witt, D. Wilkins, and D. Lenker. 1981a. Characterization of growing lettuce from density contours: I. Head selection. *Pattern Recognition* 13(**[issue no.?]**): 333-340.

_____. 1981b. Characterization of growing lettuce from density contours: II. Statistics. *Pattern Recognition* 13(**[issue no.?]**): 341-346.

Schatzki, T. F., and R. Wong. 1989. Detection of sub milligram inclusions of heavy metals in processed foods. *Food Techn.* **[volume no.?]**(**[issue no.?]**): 72-76.

Serra, J., and L. Vincent. 1989. Lecture notes in mathematical morphology. Paris, France: Ecole Nationale Superieure des Mines de Paris.

Sim, A., B. Parvin, and P. Keagy. 1996. Invariant representation and hierarchical network for inspection of nuts from x-ray images. *International Journal of Imaging Systems and Technology* 7(**[issue no.?]**): 231-237.

Tollner, E., Y. Hung, B. Upchurch, and S. Prussia. 1992. Relating x-ray absorption to density and water content in apples. *Trans. ASAE* 35(**[issue no.?]**): 1921-1928.

Viero, T., B. P. Dobrin, J. Astola, and M. Gabbouj. 1994. Further analysis of watershed algorithms. In *Proceedings of the Society for Photo-Optical Instrumentation Engineers*, , 331-342. **[what is "2300"?]**. Bellingham, Wash.: [**publisher's name?]**.

Vincent, L., and P. Soille. 1991. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *PAMI* 13(16): 583-598.